



# Revisiting content sharing in wireless ad hoc networks

Mohamed Karim Sbai, Chadi Barakat

## ► To cite this version:

Mohamed Karim Sbai, Chadi Barakat. Revisiting content sharing in wireless ad hoc networks. [Research Report] 2009. inria-00376521

**HAL Id: inria-00376521**

**<https://inria.hal.science/inria-00376521>**

Submitted on 17 Apr 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Revisiting content sharing in wireless ad hoc networks

Mohamed Karim SBAI  
EPI Planète, INRIA, France  
Email: mksbai@sophia.inria.fr

Chadi BARAKAT  
EPI Planète, INRIA, France  
Email: cbarakat@sophia.inria.fr

**Abstract**—Classical content sharing applications like BitTorrent can not be directly used in wireless ad hoc networks. When adapting them to these constrained networks, one faces two main problems. On one hand, exchanging content pieces with far nodes yields an important routing overhead. On the other hand, it is necessary to send some content pieces to far nodes to increase the diversity of information in the network, which fosters reciprocity and parallel exchanges. In this paper, we study both of these problems and propose a joint solution for them. Unlike uni-metric approaches, our solution considers relevant performance metrics together as throughput, sharing and routing overhead. We define a new neighbor selection strategy that balances between sharing and diversification efforts and decides on the optimal scope of a node. It also considers the diversification incentives problem. Through extensive simulations, we prove that our solution realizes both better download time and sharing ratio than uni-metric solutions.

## I. INTRODUCTION

The proliferation of wireless devices (Laptops, PDAs, Smartphones, etc) motivates end users to connect to each other to form spontaneous communities. A wireless multi-hop network of devices, rendered possible by the use of ad hoc routing protocols, can be a good opportunity to share some contents (data, audio, video, etc) among the members of the same community without using any established infrastructure. As the resources of a wireless ad hoc network are scarce and shared among nodes, the application used for content sharing should not rely on any central service and must divide the replication effort fairly among the members of the community while reducing the overhead on the intermediate nodes serving as relays. Considering this, file sharing applications based on the peer-to-peer (P2P) paradigm are the best candidate solutions for the following reasons. First, they have become, in a few years, the most popular applications in the Internet and users are familiar with their functionalities and features. Moreover, a P2P file sharing solution like BitTorrent [1] decentralizes the data transfer plane using the multi-sourcing concept and provides enough incentives to encourage fair sharing. It is thus mandatory to have the same principles in a wireless environment because nodes will tend to save capacity and energy. Furthermore, multi-hop wireless communications consume resources in intermediate nodes and so there is a strong need for reducing the routing overhead.

Whereas efficient content localization in wireless ad hoc

networks has attracted considerable research interest [5][6][7], the content replication problem is still in its first steps. BitTorrent [1] is the most known P2P content replication protocol that optimizes the data transfer plane. Previous studies focus on tuning BitTorrent algorithms to wireless networks to ameliorate a specific performance metric without considering all the metrics jointly. Some of them [2] aim to ameliorate the global download time by reducing the routing overhead. In fact, the proposed idea is to make peers only concentrate on their nearby neighbors. We show in this paper that if this is done, the replication burden is unequally distributed among peers and that there is a poor transmission parallelism in the network. This is contradictory to the goals of BitTorrent and is not adequate to wireless ad hoc networks. In another previous work [3][4], we propose to replicate the initial seed of the content at the edge of the network in order to increase the diversity and improve the parallelism. Although these policies register better download times and point to some new directions, they are limited to some specific cases that need to be generalized to illustrate clearly the relationship between content replication, user performance, fairness and overhead on the underlying network. On one side there is a need to diversify the content in the network to improve user perceived quality and enforce fairness, and on the other side, this diversification is costly because of the multi-hop routing. The optimal balance and the way it should be reached are still not clear.

In this paper, we study in depth the routing overhead and content diversity problems. We observe that one is facing the following dilemma: *How can one reduce the download time while having enough parallel transmissions in the network and a fair distribution of replication load ? How can one boost fair sharing by diversifying pieces of the content without increasing the routing overhead ?* Our objective is to come up with a joint solution for the routing overhead and content diversity problems. We mainly want to increase the sharing opportunities and have a minimum download time while not overloading the network. We consider in our investigation the realistic case where peers, unless they have the entire content (called leechers in the former case and seeds in the latter case), have no incentives to participate to the diversification of the content. As in BitTorrent philosophy, leechers are only interested in sharing the content with other peers having new parts of the content to reciprocate. Only seeds are generous enough to participate to the diversification of the content for

<sup>0</sup>This work has been supported by the ITEA European project on experience sharing in mobile communities (ExpeShare).

the purpose of improving global performance.

We propose a new neighbor selection strategy that distinguishes between two main efforts of peer-to-peer file sharing application: the sharing effort and the diversification effort. First, we try to answer the following question: *What must be the importance of the sharing and diversification efforts?* Our first finding is that the sharing effort must be done in a narrow area around each peer otherwise this results in an important routing overhead. We study in this work the best scope of sharing that minimizes the routing overhead. On the other side, the diversification effort is aimed to increase the entropy of information in the network to boost parallelism. That is why the diversification area should be taken wider than the sharing area. It is clear that diversification is very costly and that this effort must be done less frequently than the sharing effort. We study in this paper the impact of the diversification zone and propose an efficient strategy for scheduling sharing and diversification connections of a peer. Another important question we address is the appropriate neighbor selection strategy while the diversification effort. We prove by comparing different strategies that choosing randomly a peer in the diversification scope is the best way. Finally, and as stated before, we only implicate seed nodes in the diversification effort since they are volunteering to seed the content and so they do not wait for any return from other peers. Our solution pays attention to balance the load equally among the different seeds in a diversification area.

Using our extension of the NS-2 network simulator [9] and realistic network realizations, we prove through extensive simulations that when using our neighbor selection strategy, BitTorrent realizes both better download time and better sharing ratio than its classical Internet version. It outperforms in all regards other solutions limiting the scope of the neighborhood without diversifying pieces. We can also do better download time and better sharing ratio than when replicating the content at the edge of the network.

The remainder of this paper is organized as follows. Section II presents the background of our work and the methodology of our investigation. Section III studies the routing overhead problem and Section IV mitigates the content diversity problem. Section V presents in details our solution. Section VI summarizes our contribution and gives some future directions.

## II. BACKGROUND AND METHODOLOGY

### A. Background

- **BitTorrent** [1] is a scalable and efficient P2P content replication protocol. Each peer shares some of its upload capacity with other peers in order to increase the global system capacity. Peers cooperating together to download a content form a sharing overlay called *Torrent*. To facilitate the replication of the content in the network and to ensure multi-sourcing, a file is subdivided into a set of pieces. A peer having all pieces of the file is called a *seed*. When the peer is downloading pieces, it is called a *leecher*. Among the members of the torrent, neighbors are those with whom a peer can open a TCP connection to exchange data and information. Only four simultaneous

outgoing *active* TCP connections are allowed by the protocol. The corresponding neighbors are called effective neighbors. They are selected according to the *choking algorithm* of BitTorrent. This algorithm is executed periodically and aims at identifying the best uploaders. Once the choking period expires, a peer chooses to unchoke the 3 peers uploading to him at the highest rate. This strategy, called *tit-for-tat*, ensures reciprocity and enforces collaboration among peers. Now to discover new upload capacities, a peer chooses randomly a fourth peer to unchoke. All other neighbors are left choked. When unchoked, a peer selects a piece to download using a specific piece selection strategy. This strategy is called *local rarest first*. Indeed, when selecting a piece, a peer chooses the piece with the least redundancy in its neighborhood. Rarest first is supposed to increase the diversity of pieces[8].

- A **wireless ad hoc network** is an infrastructureless wireless network. There is no base station or central administration. Nodes of this network operate both as routers and hosts. The determination of the next hop for a packet is solved by the ad hoc routing protocol (e.g. DSDV [10]) which ensures optimal path computation in a distributed and adaptive way. Unlike in wired networks, the resources of a wireless ad hoc network are scarce and shared among nodes. When an end-user sends a data packet, some other end-users necessarily consume from their capacities and energies to deliver it to the destination. Other end-users suffer from the transmission interference.

### B. Scenario and Methodology

We start from the BitTorrent protocol and consider the interesting case where all nodes of a wireless ad hoc network are interested in sharing the same content. We want to understand the performance of BitTorrent in this challenging dense case before moving to more sparse torrents. Indeed, when the torrent is dense, the routing overhead on peers is the maximum since on one hand the volume of exchanged data is large, and on the other hand any packet sent over multiple hops will steal bandwidth from all intermediate nodes which are also peers. Note that a packet relayed by a node at the routing layer is not seen by the applications running in this node, in particular the BitTorrent application. Furthermore, we consider that the nodes are fixed and randomly distributed in the plane in such a way to form a big connected network. This means that content diversity cannot be obtained without sending data to far away nodes. We leave the study of a mobile scenario for a future research, but note for now that if nodes were mobile, their movements would help in increasing the entropy of information in the network, making the scenario less challenging. Lastly, we consider that nodes have access to a routing table, which can be provided by a proactive ad hoc routing protocol (e.g. DSDV).

In our investigation, we proceed with an experimental approach using the NS-2 network simulator [9]. In fact, we extend NS-2 by implementing a tunable BitTorrent-like module allowing content sharing in wireless ad hoc networks. Mainly, one can change the neighbor selection and piece selection strategies of the BitTorrent client and measure

the resulting performance. In addition to the data transfer plane, our module implements a peer discovery mechanism on each peer. This mechanism emulates for the BitTorrent client the existence of a centralized tracker providing it with the list of torrent members. As we focus in this work on the data transfer plane, the optimization of the membership management mechanism is out of the scope of this paper. Furthermore, our module profits from the existing NS-2 modules to ensure wireless communication and multi-hop routing of packets. Figure 1 gives an idea of the per-node layered architecture of the system that we are simulating.

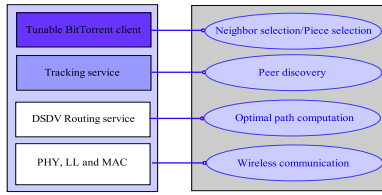


Fig. 1. Per-node architecture

The wireless ad hoc network that we are simulating consists of 50 nodes randomly distributed in a  $500m \times 80m$  square area. In our simulations, we discard all the realisations where the topology is not connected. Nodes connect to each other using the 802.11 MAC Layer with the RTS/CTS-Data/ACK mechanism enabled. The data rate is set to 11 Mb/s and the wireless range to 50m. The ad hoc routing service is ensured thanks to the DSDV proactive protocol [10]. At the beginning of each simulation, a random node is chosen as the seed of the content and the other nodes are leechers. The content is a 100 Mbytes data file that is subdivided into 1000 pieces. All peers start downloading the file at the same time (a flash crowd scenario). The BitTorrent choking period is set to 40s.

### III. THE ROUTING OVERHEAD PROBLEM

The first question we address is about the optimal neighbor selection strategy for P2P content sharing in wireless ad hoc networks. We start in our investigation from the classical version of BitTorrent and we vary the scope of the sharing area of the peers. This scope represents the maximum number of hops between peers authorized to exchange pieces. The version deployed in the Internet, called later classical version for short, corresponds to a sharing scope equal to the maximum number of hops in the network.

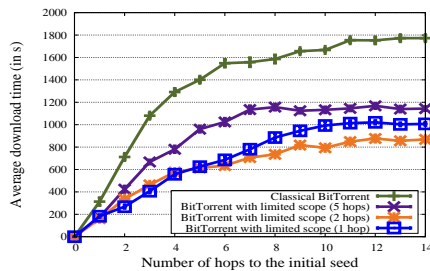


Fig. 2. Download time per peer vs. number of hops to the initial seed

Figure 2 plots the average download time of the content per peer as a function of the number of hops of that peer to the initial seed node for different values of the sharing

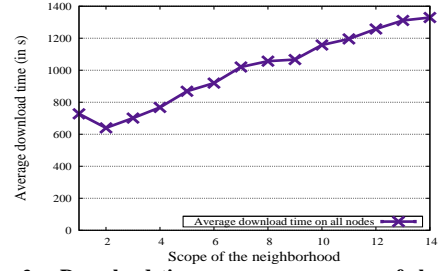


Fig. 3. Download time per peer vs. scope of sharing

scope. On one hand, one can notice that the download time increases with the number of hops to the seed for all values of the sharing scope. This is mainly due to the fact that the achievable throughput of TCP decreases considerably with the number of hops in a wireless ad hoc network. Peers far from the seed got most if not all of the pieces of the content in multi-hop via other peers. On the other hand, the classical version of BitTorrent has the highest download time, because of the routing overhead and the degradation of TCP performance in a multi-hop environment. One needs to limit the sharing scope to a small value, say for example one or two, to obtain the best performances. Figure 3 consolidates this observation by showing the average download time over all nodes as a function of the scope of sharing. This figure shows an amelioration in the download time when this sharing scope is reduced. In fact, the routing overhead is minimal when the scope is the smaller, otherwise pieces of content will be forwarded by intermediate nodes at the routing layer without profiting from them at the BitTorrent layer, which incurs lot of overhead on these nodes. Moreover, additional transmissions are needed later on to send the same pieces to these intermediate nodes. Another important factor is that the throughput of TCP is better over short distances and in this case, more pieces of data can be sent during the choking time slot. One can see in the figures a slightly better gain for a scope of two hops compared to a scope of one hop even though there is some routing overhead in the former case. In fact, the two hop case allows more neighbors, which leads to a better sharing and a better forwarding of pieces. At the same, the routing overhead is still small so that we can notice an overall gain. Unfortunately, for scopes larger than two hops, the routing overhead becomes important to kill any gain from having more neighbors.

### IV. THE PIECE DIVERSITY PROBLEM

In the previous section, we have shown that decreasing the scope of sharing ameliorates the download time. In this paragraph, we try to answer the following questions: *By limiting the scope of sharing, are we limiting the sharing opportunities between nodes ? Is there fair sharing among them in this case ? Is there a piece diversity problem ?*

When the sharing scope is very limited, the pieces of the content will mostly propagate from the initial seed to the farthest nodes in a unique direction via other nodes in between. Far nodes do not have original pieces to provide

to upstream nodes that are closer to the initial seed. That is why, nodes fail to reciprocate data with each other, and hence, the load of sharing is not equally divided among them. In general, the farther the nodes are from the initial seed the less packets they will have to send. Moreover, there will be no diversity of pieces in the network. The same pieces will propagate from one neighborhood to another, which can not result in a fair exchange.

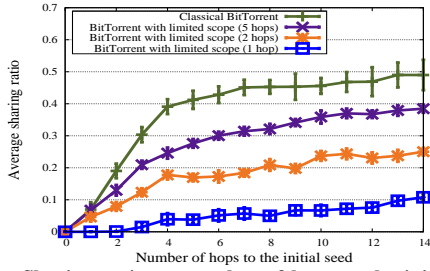


Fig. 4. Sharing ratio vs. number of hops to the initial seed

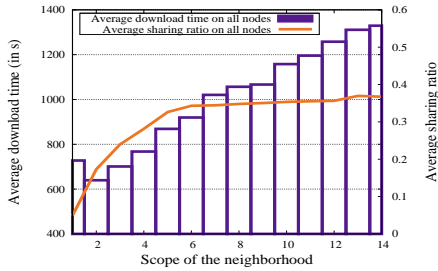


Fig. 5. Sharing ratio vs. number of hops to the initial seed

To strengthen this claim, we plot in Figure 4 the sharing ratio as a function of the number of hops to the initial seed for the same simulations used in the previous section. The sharing ratio between a couple  $(i, j)$  of peers is defined as:  $R_{ij} = \frac{\min(D_{ij}, D_{ji})}{\max(D_{ij}, D_{ji})}$ ; where  $D_{ij}$  is the amount of data that node  $i$  has downloaded from node  $j$  during the torrent lifetime. This ratio measures the magnitude of the reciprocity between two nodes. A value nearing null means a one-way propagation of data. The fair sharing ideal case is obtained when the sharing ratio is equal to 1. From Figure 4, one can observe the following:

- The sharing ratio increases slightly with the number of hops to the initial seed. In fact, far peers can reciprocate some data with other far peers having almost the same distance to the seed because they can both receive some different pieces on different paths. The farthest the nodes are from the seed the more of these different paths exist. But the diversity of pieces is still low as the dissemination is done in a unique direction.
- When one decreases the scope of sharing, the sharing ratio decreases dramatically. The cases where the scope is set to one hop or 2 hops yield the lowest sharing ratios, which can be explained by the fact that pieces propagate like a wave from the initial seed to the farthest nodes. The resources of the network are not fully used since nodes wait for pieces to arrive to their neighborhood and rarely have original pieces to provide to their neighbors. The P2P sharing application behaves then like a simple piece relaying protocol that ignores parallel transmission abilities of the network and the distribution of the load among peers.

- For large scopes of sharing, for instance the classical BitTorrent case, the sharing ratio is still lower than  $\frac{1}{2}$  because the routing overhead is important and the number of pieces that can be downloaded during a choking slot is very small, mainly when the number of hops between neighbors is large. So, there is a trade-off between diminishing the routing overhead and increasing piece diversity in the network. Figure 5 plots both the sharing ratio and download time averaged over all nodes vs. the scope of sharing. From sharing perspectives, one can notice that it is useless in our settings to increase the scope beyond 5 because of the degradation of TCP throughput with the number of hops. From download time perspectives, the best is to limit the scope to a very small value as two hops to limit the routing overhead. *Can one do better ?* In the following section, we prove that this is possible by decoupling the sharing effort from the diversification effort. Mainly we introduce a new algorithm to increase the diversity of pieces in the network at a limited routing cost, and in parallel we limit the sharing scope to two hops in order not to overload the network. This way we can do better than the simple small scope case by having better diversity of pieces and hence more parallel transmissions and better reciprocity. At the same time, we are better than the simple large scope case in terms of sharing because we can diversify pieces in the network to improve sharing, but without suffering from the routing overhead problem.

## V. SOLVING THE DILEMMA

In the two previous sections, we presented two problems related to content sharing in wireless ad hoc networks. We are facing the following dilemma: On one hand, decreasing the scope of sharing ameliorates the download time but it results in a very weak parallelism in the network due to a lack in piece diversity. On the other hand, increasing the scope of sharing increases the diversity of pieces in the network, but at the cost of more routing overhead which yields to worse download time. In this paragraph, we present our solution to this dilemma. Our objective is to come up with a joint solution for the routing overhead and piece diversity problems. In designing our new neighbor selection strategy, we took into consideration the following points:

- Data transfers between distant peers suffer from very poor performances in wireless ad hoc networks. Hence, a leecher has no incentives to send pieces of the content to far nodes as they will not be able to serve him back with a high throughput. Moreover, leechers that are far from the initial seed have less original pieces to provide them to nearer leechers. Considering this, we decided that in our neighbor selection strategy, only seeds send pieces to far peers. Indeed, a seed is a volunteer peer that serves others without expecting any return from them. The leechers have more incentives to concentrate on peers located in their close neighborhood provided that there are original pieces to share with them.
- If all seeds send pieces to far nodes at the same time, the routing overhead will be again important and we will loose in performance. In our solution, we subdivide the piece diversification effort among seeds both in space and time.

- If a seed can not send a complete piece to the selected peer during the choking slot, the gain in diversity will be null since the smallest unit that a peer can share with others is the piece. In our solution, we limit the scope of diversification of seeds to the number of hops allowing the transfer of a complete piece. Pieces are spread in other parts of the network by other peers becoming seeds and deciding to stay in the torrent.

#### A. The neighbor selection strategy

- In the **leecher state**, peers concentrate on their nearby neighborhood. The scope of sharing is fixed to 2 hops as it is proved to be the best regarding the routing overhead and transfer performance. A leecher maintains 4 simultaneous active outgoing connections. The 3 first connections are dedicated to best uploaders among peers in the sharing scope and the fourth connection consists in an optimistic unchoke allowing to discover new upload capacities and the bootstrap of the sharing. The fourth peer is chosen randomly among leechers within the sharing scope. The selection is done at the end of each choking time slot. Except the limitation of the scope to two hops, this is globally the classical BitTorrent algorithm for leechers.
- In the **seed state**, peers dedicate their 3 first connections to serve leechers within their sharing scopes (set to 2). These are the connections dedicated to inject the content in the network by starting from the small sharing area. The fourth connection of a seed is mainly dedicated to diversify pieces in an area wider than the sharing one. This area is called the diversification area of the seed and contains all peers not belonging to its sharing area and that are located at a distance lower than the diversification scope. The scope of diversification is determined by observing the range of piece transmissions (see paragraph V-D). Paragraph V-C studies the optimal way to choose a leecher in the diversification area and paragraph V-B shows how the fourth connection is used when there is more than one seed within the diversification area.

#### B. Dividing the diversification effort among seeds

Sending pieces to far nodes engenders an important routing overhead. Hence, the diversification effort must be divided between all seeds both in space and in time. In our solution, each seed is responsible of its own diversification area. Thus, the network is subdivided into regions and a single seed does not have to serve the whole network. Moreover, when there are many seeds within the diversification scope of each other (for example when other peers finish the download and decide to stay), our solution reduces the routing overhead of the fourth connection of each of them dedicated to diversification by the number of seeds in its diversification area. This is done as follows. The seed pauses for a number of slots equal to the number of seeds in its diversification area between every two diversification time slots. During the pause, the seed can serve leechers in its sharing area. This scheduling is repeated periodically and follows the evolution of the number of seeds. In this way, the total diversification overhead is kept constant as there are more and more seeds in the network.

#### C. Optimal diversification-neighbor selection strategy

In this paragraph, we look for the best strategy used by seeds to select leechers in their diversification areas at time slots. The goal is to maximize the sharing ratio while minimizing the average download time. Let's note the sharing scope of a seed as  $S_s$  and its diversification scope as  $S_d$ . Searching the optimal strategy, we define a parametric general probability distribution to tune the leecher selection and we study, through simulations and by varying the parameter of the distribution, the impact of the different strategies on the torrent performances. Indeed, we model the probability to select a peer located at  $h$  hops from a seed in its diversification scope ( $S_s < h \leq S_d$ ) as follows:

$$p(h) = \frac{h^\alpha}{\sum_{l=S_s+1}^{S_d} N_l l^\alpha} \quad (1)$$

where  $N_l$  is the number of peers located at  $l$  hops and  $\alpha$  is a parameter of the probability distribution. The sum of this probability function over all peers in a diversification area is clearly equal to 1. By setting  $\alpha$  to 0, one can obtain the uniform probability distribution where peers are selected with the same probability independently of their location. For large positive values of  $\alpha$ , the probability to select the farthest peers becomes close to 1 and that to select peers near to the seed almost null. For large negative values of  $\alpha$ , one gets the opposite case; the seed diversifies pieces over peers close to it. This parameter  $\alpha$  covers then a large set of strategies and its optimal value should point us to the optimal leecher selection strategy to use for diversification purposes. We seek this optimal value next by the help of extensive simulations.

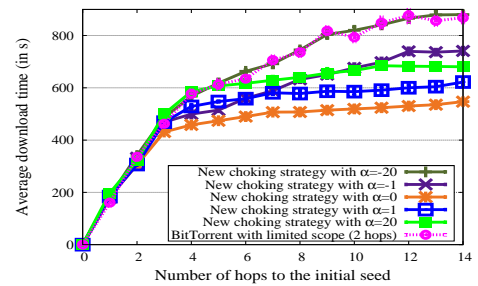


Fig. 6. Download time vs. number of hops to the initial seed

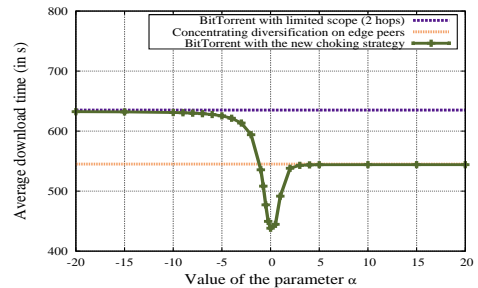


Fig. 7. Download time vs.  $\alpha$

Figure 6 plots the average download time as function of the number of hops to the initial seed for different values of the



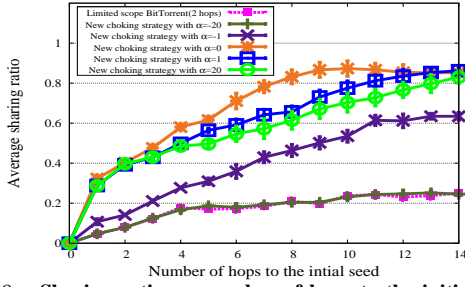


Fig. 8. Sharing ratio vs. number of hops to the initial seed

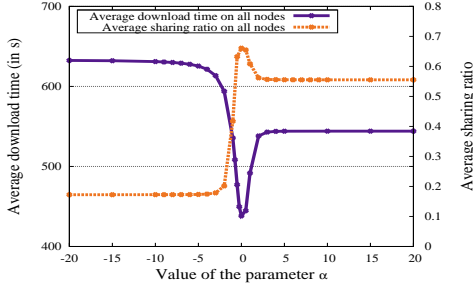


Fig. 9. Sharing ratio and download time vs.  $\alpha$

parameter  $\alpha$ . Figure 7 draws the average download time over all peers as a function of the parameter  $\alpha$ . For large negative values of  $\alpha$ , the download time is maximal and tends to the one obtained without diversifying the pieces (scope 2 in Figure 5). For large positive values of  $\alpha$ , one can obtain a better performance since there is the introduction of some diversity of pieces in the network but the concentration is only on leechers located at the edge of the diversification area. This is below the optimal because of the routing overhead and an inefficient spatial distribution of pieces. Our main observation is that a value of  $\alpha$  equal to 0 gives the best performance. Indeed, for this optimal value, seeds distribute pieces uniformly in the network and then boosts fair sharing among peers and transmission parallelism while having a reasonable average routing overhead. Figure 8 plots the sharing ratio as a function of the number of hops to the initial seed for different values of the parameter  $\alpha$ . Figure 9 presents both the average sharing ratio and the average download time over all nodes as function of the parameter  $\alpha$ . These figures prove that the best choice of  $\alpha$  is 0 as it also results in the best sharing ratio. Indeed, large negative values of  $\alpha$  means no diversification of pieces and then the lowest sharing ratio is recorded, whereas for large positive values, one can obtain a better sharing ratio but the load is not equally divided among leechers. For the uniform probability case, the sharing opportunities are the best because original pieces are distributed over all peers and sharing areas and the load is not concentrated on any local neighborhood.

#### D. Choice of the diversification scope

The results shown till now are obtained for a diversification scope set to 10. In this paragraph, we study the impact of this scope both on download time and sharing ratio. Figure 10 plots the average download time and the average sharing ratio over all nodes Vs. the scope of diversification. It shows that for small values of this scope, there is not enough diversity introduced into the network. Hence, the sharing ratio decreases

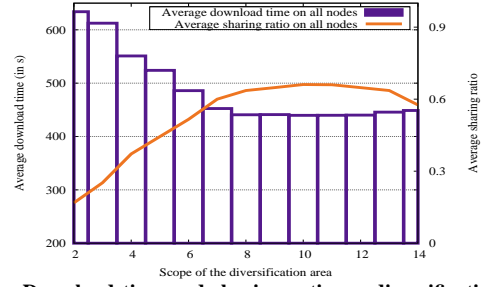


Fig. 10. Download time and sharing ratio vs. diversification scope

considerably and the download time worsens. For large diversification scopes, again the sharing ratio and download time worsen for the simple reason that TCP becomes unable to send entire reusable pieces far away from the seed. It is clear that in our settings a diversification scope around 10 hops away from each seed leads to best performances both from download time and sharing ratio perspectives. This should be the largest scope where entire content piece could be sent. As a result, the diversification scope must be fixed so that it does not exceed the range of pieces. To support this in practice, the seed can adapt its diversification scope up or down by measuring the number of pieces that it can send during a choking slot to leechers located at the edge of the diversification area. We leave this online adaptation for a future research.

## VI. CONCLUSIONS AND PERSPECTIVES

In this paper, we study the routing overhead and piece diversity dilemma and propose an efficient neighbor selection strategy that minimizes download time while maximizing sharing opportunities. Our proposed choke/unchoke algorithm is practical and does not make any assumption on the cooperation of leechers. We prove through extensive NS-2 simulations that when using our neighbor selection strategy, the download time of BitTorrent is 30% of the one obtained with the classical version of BitTorrent and 65% of the one obtained with the version without diversification of pieces and a sharing scope equal to 2. As for the sharing, the load is better distributed among peers in our version, and so the average sharing ratio easily exceeds  $\frac{1}{2}$ . However, the solution and results presented in this paper are those for the dense challenging scenario. We study sparse and mobile P2P networks in our future work.

## REFERENCES

- [1] BitTorrent protocol. <http://wiki.theory.org/BitTorrentSpecification>.
- [2] Michiardi P., Urvoy-Keller G., "Performance analysis of cooperative content distribution for wireless ad hoc networks", WONS 07.
- [3] M.K.Sbai, C.Barakat, J.Choi, A.Al Hamra, T.Turletti, "Adapting BitTorrent to wireless ad hoc networks", Ad-Hoc Now 08.
- [4] A.AlHamra, C.Barakat, T.Turletti, "Network Coding for Wireless Mesh Networks: A Case Study", WoWMoM 06.
- [5] A.Klemm, C.Lindermann, O.Waldhorst, "A special-purpose peer-to-peer file sharing system for mobile ad hoc networks", VTC 2003.
- [6] S.M. Das, H.Pucha, Y.C. Hu. Ekta: an efficient peer-to-peer substrate for distributed applications in mobile ad hoc networks., Purdue 04.
- [7] L.B. Oliveira, I.G.Siqueira, A.A.Loureiro, "Evaluation of ad hoc routing protocols under a peer-to-peer application", WCNC 03.
- [8] A. Legout, G. Urvoy-Keller, P. Michiardi, "Rarest First and Choke Algorithms Are Enough", IMC 06.
- [9] NS: The Network Simulator, <http://www.isi.edu/nsnam/ns/>
- [10] C.E. Perkins, P.Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers", SIGCOMM 94.